CPROG08SZU/M

March 2022

CPROG08SZ USER GUIDE

Purchase Agreement

P&E Microcomputer Systems, Inc. reserves the right to make changes without further notice to any products herein to improve reliability, function, or design. P&E Microcomputer Systems, Inc. does not assume any liability arising out of the application or use of any product or circuit described herein.

This software and accompanying documentation are protected by United States Copyright law and also by International Treaty provisions. Any use of this software in violation of copyright law or the terms of this agreement will be prosecuted.

All the software described in this document is copyrighted by P&E Microcomputer Systems, Inc. Copyright notices have been included in the software.

P&E Microcomputer Systems authorizes you to make archival copies of the software and documentation for the sole purpose of back-up and protecting your investment from loss. Under no circumstances may you copy this software or documentation for the purpose of distribution to others. Under no conditions may you remove the copyright notices from this software or documentation.

This software may be used by one person on as many computers as that person uses, provided that the software is never used on two computers at the same time. P&E expects that group programming projects making use of this software will purchase a copy of the software and documentation for each user in the group. Contact P&E for volume discounts and site licensing agreements.

P&E Microcomputer Systems does not assume any liability for the use of this software beyond the original purchase price of the software. In no event will P&E Microcomputer Systems be liable for additional damages, including any lost profits, lost savings or other incidental or consequential damages arising out of the use or inability to use these programs, even if P&E Microcomputer Systems has been advised of the possibility of such damage.

By using this software, you accept the terms of this agreement.

Manual version 1.03

CPROG08SZ USER GUIDE

1 INTRODUCTION

CPROG08SZ is a command-line version of the PROG08SZ software which programs EEPROM, EPROM, Flash, etc. either in-circuit or socketed in an NXP (orig. Motorola) ICS. PEmicro offers two hardware interfaces for MON08 targets, and produced other interfaces which have since been discontinued. The discontinued items are considered legacy interfaces and are no longer actively supported.

The interfaces are organized into various classes, and this class must be specified when using CPROG08SZ. Cyclone Universal LC and Cyclone Universal FX are considered Class 8, as is the legacy Cyclone PRO interface. The legacy MON08 Multilink was a parallel port interface and is Class 7. The legacy MON08 Cyclone is Class 5. CPROG08SZ will work with these Class 1-8 devices.

2 STARTUP COMMAND-LINE PARAMETERS

Start the programming software by clicking it's icon, or running it from the MSDOS prompt. You may have to edit the ICON to add the name of the configuration (script) file to the command line. Allowable command line parameters are:

[?] Use the '?' character option to cause the command-line programmer to wait and display the result of programming in the PROG window. If the user does not use a batch file to test errorlevel, this provides a method to display the programming result. This option should be the FIRST commandline option.

[filename] A configuration file containing programming commands and comments (default = prog.cfg). See **Section 3**

CONFIGURATION (SCRIPT) FILE for more information.

[v] V causes the programmer not to check the range of S-record addresses before programming or verifying. This speeds up the programming process. The option should beused with care as all out of range s-records will be ignored.

3 CONFIGURATION (SCRIPT) FILE

The configuration file contains two types of commands. The first type are SETUP commands, which configure the parameters that are used when attempting to contact the target (Detailed in **Section 5 SETUP COMMANDS FOR STARTUP**). These SETUP commands include choosing the proper COM port, delays, and target class to use when contacting the target. All SETUP commands begin with a colon (:). The entire configuration file is parsed prior to contacting the target in order to properly process all SETUP commands. After this is complete, the CPROG08SZ software processes the configuration file for programming commands (detailed in **Section 6 PROGRAMMING COMMANDS**). These commands are used to control erasure, programming, and verification of target device programming. All programming commands consist of two uppercase letters, possibly followed by parameters according to the particular command.

4 PASSING SECURITY

By default, when the programmer starts up, it attempts to establish communications with the target without passing security (this can be changed as specified in **Section 5 SETUP COMMANDS FOR STARTUP**). When an EM (Erase Module) command is executed, the programmer will first erase the target and then attempt to pass security. The idea behind this is that the default configuration file will almost immediately attempt to erase the target device, hence passing security is easier because the programmer knows the security code for an erased device. With this default configuration, the first two programming commands (after all SETUP commands processed) should be CM (Choose Module) and EM (Erase Module). The user can elect, using SETUP commands, to attempt to pass security on startup by specifying the :FORCEPASS and :SECURITYCODE SETUP commands in the configuration file.

5 SETUP COMMANDS FOR STARTUP

SETUP commands are all processed before the programmer attempts to contact the target. The whole configuration file is parsed for these commands prior to attempting communications. This section gives an overview of using these SETUP commands to do different type of configuration. The default base for parameters is decimal, with the exception of the :SECURITYCODE command. An overview of the SETUP commands is as follows:

:PORT n

1..8 - Specifies appropriate serial or parallel interface to use. Class 1-7 only.

USB1..USB8 - Use appropriate USB port. Class 7-8 only.

COM1..COM8 - Use appropriate serial port.

#.#.#.# - IP address for Ethernet communications. Class 8 only.

:TARGETCLASS n Sets the target class to:

1 - Motorola ICS Board with processor installed.

2 - Motorola ICS Board without processor.

3 - Serial port direct to target w/with DB9 connection.

4 - Serial port direct to target w/with DB9 connection and auto-reset circuit built in.

5 - PEmicro MON08-Cyclone (legacy) connected to target via MON08.

6 - PEmicro POWER08 Interface (legacy) to target with DB9 connection.

7 - PEmicro MON08 Multilink (legacy) connected to target via MON08.

8 - PEmicro Cyclone Universal LC, Cyclone Universal FX, Cyclone PRO (legacy) connected to target via MON08.

:BAUD n Set the baud rate to n. This is for class 1-4 only. Class 5, 6, 7, and 8 use autobaud detection.

:POWERDOWNDELAY n

Amount of time to delay when the power to the target is turned off for the targets power supply to drop to below 0.1v. n is the time in milliseconds.

:POWEROFFONEXIT n

Determines whether power to the target should be turned off when the CPROG08SZ application terminates (for class 5, 6, 7, and 8 only). Valid values of n are:

0 : Turn power off upon exit (default)

1 : Keep power on upon exit

:POWERUPDELAY n

Amount of time to delay when the power to the target is turned on OR the target is reset, and before the software attempts to talk to the target. This time can be a combination of power-on time and reset time (especially if a reset driver is used). n is the time in milliseconds.

:IRQ n Class 5, 7, and 8 targets only. Sets the value of the IRQ pin on reset. This pin can either be set to high voltage (n is 1) or to ground (n is 0).

:FORCEPASS Specified that security should be passed on startup of the software instead of waiting for an EM (Erase Module) command. The :SECURITY code command must also be provided.

:SECURITYCODE hhhhhhhhhhhhhhhh

Specifies the 8 bytes of security code to use at startup which corresponds to the addresses $FFF6-$FFFD of the target HC08 device. The parameter for this is a string containing 8 bytes of data in HEX separated by white spaces.

:DEVICETYPE str For Class 5, 6, 7, and 8 only. Specifies the target device family. As an example, the device type for a 68HC908KX8 would be KX. The allowed device type values are :

AB, AP, AS, AT, AZ, BD, EY, GP, GR16, GR4/8, GT, GZ, JB16, JB1/8, JG, JK, JL, KX, LD, LJ, MR4/8, MR16/32, QT, QY, RF, RK, SR

:DEVICECLOCK n For Class 5, 6, 7, and 8 only. Controls whether the PEmicro interface should drive a clock to the target or whether the PEmicro interface should tristate its clock output. Valid values of n are:

0 : Clock driven by PEmicro Interface

1 : Target self-clocked, PEmicro Clock output disabled

:CLOCKDIVIDER n For Class 5, 6, 7, and 8 only. Often one of the port pins of the target processor controls the ratio of the BUS clock to the External clock. Valid values of n are:

0 : Divide by 2 (usually and if applicable)

1 : Divide by 4 (usually and if applicable)

:DEVICEPOWER n For Class 5, 6, 7, and 8 only. This setting incorporates both the target voltage as well as whether the PEmicro interface is generating/switching the power or whether the user should be prompted to cycle the target power. Valid values of n are:

0 : 5 Volts, Generated/Switched by PEmicro Interface

1 : 5 Volts, User Switched (when prompted)

2 : 3 Volts, Generated/Switched by PEmicro Interface

3 : 3 Volts, User Switched (when prompted)

4 : 2 Volts, Generated/Switched by PEmicro Interface

5 : 2 Volts, User Switched (when prompted)

6 PROGRAMMING COMMANDS

Commands all start with a two character sequence, followed by white space (blanks or tabs). Lines starting with characters that are not commands are listed as REMarks. The values for starting, ending, byte, word, user_par and base_address are given in hexadecimal. The term *filename* refers to the full path to a file. Commands use the same two letter codes as used in the interactive programmer PROG08SZ. The same .08P files used by PROG08SZ are used to set up a particular device to be programmed. If a user function is specified for a particular device, its two character command and the meaning or user_par are specified in the .08P file.

BM	Blank check module.
BR *starting ending*	Blank check range.
EB *starting ending*	Erase byte range.
EW *starting ending*	Erase word range.
EM	Erase module.
PB *starting byte ... byte*	Program bytes.
PW *starting word ... word*	Program words.

PM	Program module.
CM *filename base_address*	Choose module .08P file.
VM	Verify module.
VR *starting ending*	Verify range.
VC	Verify module CRC-8.
UM *filename*	Upload module.
UR *starting ending filename*	Upload range.
SS *filename*	Specify S record.
SM *starting ending*	Show module.
QU	Quit.
RE	Reset chip.
GO	Start Device Running (optionally use as last command if you want the device to run for testing. Should be immediately preceded with an 'RE' command).
DE *timeinms*	Delays "timeinms" milliseconds
xx *user_par*	Only for user function specified in .08P file.

Example User functions: - User functions are listed in the .08P file

PT	"PT" Program Trim function available in the QT or QY algorithms.
E1	"E1" Erase EEPROM 1 Module

7 EXAMPLE PROGRAMMING SCRIPT FOR CLASS 8 INTERFACE -> HC08QY4 TARGET

EXAMPLE.CFG

```
:DEVICEPOWER 0
:POWERDOWNDELAY 250
:POWERUPDELAY   250
:POWEROFFONEXIT 0
```

```
:PORT USB1
:TARGETCLASS 8
:DEVICETYPE QY
:CLOCKDIVIDER 0
:DEVICECLOCK 0
:OUTPUTCLOCK 1
CM 908_QY4.08P
SS HC08QY4.s19
EN  ;Erase if not Blank
PM  ;Program Module
VC  ;Verify Checksum
VM  ;Verify Module
```

8 EXAMPLE PROGRAMMING SCRIPT FILE FOR PEmicro MON08 CYCLONE -> 908SR12 TARGET

The programming script file should be a pure ASCII file with one command per line.

An example is:

```
; Example Configuration for in-circuit programming a 68HC908SR12 device
; using a PEmicro MON08 Cyclone.
;
; Pinouts of the SR MON08 target header from the manual:
;
;           1      2
;         NC *  * GND
;         NC *  * NC
;         NC *  * IRQ
;         NC *  * NC
;         NC *  * PTA0
;         NC *  * PTA2
;  OSCout *  * PTA1
;       Vout *  * PTC1
;           15       16
;

:PORT           1
:TARGETCLASS    5       ; PEmicro MON08 Cyclone Interface
:POWERDOWNDELAY 500
:POWERUPDELAY   500
:DEVICETYPE     SR      ; SR Device. See manual for header.
:DEVICECLOCK    0       ; OSCout Clock driven by PEmicro Interface
:CLOCKDIVIDER   1       ; Bus is XTAL Divide by 4
:DEVICEPOWER    0       ; MCU voltage is 5 Volts, and is
                                     ; Generated/Switched by the MON08 Cyclone

CM C:\PEMICRO\PROG08SZ\908_SR12_highspeed.08P                    ; Choose Flash
                                                                  Algorithm
EM ; Erase Flash Module
BM ; Blank Check Flash Module
SS C:\PEMICRO\PROG08SZ\TESTSR12.S19 ; Specify S-record for program
PM ; Program Flash Module
VC ; Verify programmed data CRC-8
```

9 EXAMPLE PROGRAMMING SCRIPT FILE FOR PEmicro MON08 MULITILINK -> 908GP32 TARGET

The programming script file should be a pure ASCII file with one command per line.

An example is:

```
; Example Configuration for in-circuit programming a 68HC908GP32 device
; using a PEmicro MON08 Multilink.
;
; Pinouts of the GP MON08 target header from the manual:
;
;          1       2
;        NC *  * GND
;        NC *  * RESET
;        NC *  * IRQ
;        NC *  * PTA0
;        NC *  * PTA7
;        NC *  * PTC0
; OSCout *  * PTC1
;      Vout *  * PTC3
;          15       16
;

:PORT            1       ; Parallel Port 1
:TARGETCLASS    7       ; PEmicro MON08 Multilink Interface
:POWERDOWNDELAY 500
:POWERUPDELAY   500
:DEVICETYPE     GP      ; GP Device. See manual for header.
:DEVICECLOCK    0       ; OSCout Clock driven by PEmicro Interface
:CLOCKDIVIDER   1       ; Bus is XTAL Divide by 4
:DEVICEPOWER    0       ; MCU voltage is 5 Volts, and is
                                        ; Generated/Switched by the MON08 Multilink

CM C:\PEMICRO\PROG08SZ\908_GP32_highspeed.08P
EM
BM
SS C:\PEMICRO\PROG08SZ\TESTGP32.S19
PM
VM
```

10 EXAMPLE PROGRAMMING SCRIPT FILE FOR ICS PROGRAMMING OF SOCKETED DEVICES

The programming script file should be a pure ASCII file with one command per line.

An example is:

```
:PORT        1
:POWERDOWNDELAY 500
:POWERUPDELAY   500
:TARGETCLASS    1       ; Socket device in ICS
```

```
:BAUD 9600

CM C:\PEMICRO\PROG08SZ\908_GP32_highspeed.08P
EM
BM
SS C:\PEMICRO\PROG08SZ\TESTGP32.S19
PM
VM
```

11 SAMPLE BATCH FILE

Here is an example of a batch file that will call the command-line programmer and test its error code return.

Windows 7/8/10/11:

```
C:\PEMICRO\PROG08SZ\CPROG08SZ
C:\PEMICRO\PROG08SZ\ENGINE.CFG
if errorlevel 1 goto bad
goto good
:bad
ECHO BAD BAD BAD BAD BAD BAD BAD BAD
:good
ECHO done
```

12 ERROR RETURNS

Error returns are provided so they may be tested in .BAT files. The error codes used are:

0 - Program completed with no errors.

1 - Cancelled by user.

2 - Error reading S record file.

3 - Verify error.

4 - Verify cancelled by user.

5 - S record file is not selected.

6 - Starting address is not in module.

7 - Ending address is not in module or is less than starting address.

8 - Unable to open file for uploading.

9 - File write error during upload.

10 - Upload cancelled by user.

11 - Error opening .08P file.

12 - Error reading .08P file.

13 - Device did not initialize.

14 - Error loading .08P file.

15 - Error enabling module just selected.

16 - Specified S record file not found.

17 - Insufficint buffer space specified by .08P to hold a file S record.

18 - Error during programming.

19 - Start address does not point into module.

20 - Error during last byte programming.

21 - Programming address no longer in module.

22 - Start address is not on an aligned word boundary.

23 - Error during last word programming.

24 - Module could not be erased.

25 - Module word not erased.

26 - Selected .08P file does not implement byte checking.

27 - Module byte not erased.

28 - Word erase starting address must be even.

29 - Word erase ending address must be even.

30 - User parameter is not in the range.

31 - Error during .08P specified function.

32 - Specified parallel printer port is not available.

33 - Command is inactive for this .08P file.

34 - Cannot enter background mode. Check connnections.

35 - Not able to access processor. Try a software reset.

36 - Invalid .08P file.

37 - Not able to access processor RAM. Try a software reset.

38 - Initialization cancelled by user.

39 - Error converting hexadecimal command number.

40 - Configuration file not specified and file prog.cfg does not exist.

41 - .08P file does not exist.

42 - Error in io_delay number on command line.

43 - Can not talk to MON08 interface.

44 - Error specifying decimal delay in milliseconds.

45 - Can not talk to MON08 interface.

46 - Error. You must erase a secured device before you can program it.

47 - Error in script file.

48 - Error contacting target.

49 - Error communicating to MON08 interface board.

50 - S-Record file does not contain valid data.

51 - CRC-8 Verification failure - S-record data does not match MCU memory.

52 - Sorting must be enabled to verify flash checksum.